Discrete Applied Mathematics 15 (1986) 241-248
North-Holland

MONGE SEQUENCES AND A SIMPLE ASSIGNMENT ALGORITHM*

U. DERIGS, O. GOECKE** and R. SCHRADER

Institut für Ökonometrie und Operations Research, Abteilung Operations Research, Rheinische Friedrich-Wilhelms-Universität Bonn, Nassestraße 2, D-5300 Bonn 1, West Germany

Received 16 April 1986

In 1781 the French mathematician G. Monge gave an optimality criterion and a greedy-like procedure for solving specially structured transportation problems. Here we present a 'new' approach for solving assignment problems which builts upon Monge's observation. The basic idea is to transform an arbitrary assignment problem into an 'equivalent' one where Monge's condition is fulfilled.

1. Introduction

The assignment problem is one of the celebrated (optimization) problems in graph theory, combinatorial optimization and linear programming. It arises as a special case (structure) within more general concepts like network flow, matroid intersection and general matching. This makes this problem genuine and attractive especially from a theoretical and algorithmic point of view and quite a number of 'different' assignment algorithms have been developed in the literature throughout the last thirty years. Why then present another assignment algorithm?

Before we answer this question by stating our intentions let us first introduce the problem formally.

For our purpose the following graph-theoretic formulation turns out to be most adequate:

Assume $G=(V_1, V_2, E)$ is a complete bipartite graph where w.l.o.g. $V_1=V_2=\{1,\ldots,n\}$ and let $C=(c_{ij})$ be a $n\times n$ cost-matrix. For $F\subseteq E$ we define

$$c(F):=\sum_{(i,j)\in F} c_{ij}.$$

A (*perfect*) *matching* in G is a subset $M\subseteq E$ such that every node is incident to at most one (exactly one) edge in M. Perfect matchings are also called *assignments* in G. Then the assignment problem (AP) is defined as follows:

*Supported by Sonderforschungsbereich 21 (DFG), Institut für Ökonometrie und Operations Research, Universität Bonn.

**Supported by Studienstiftung des deutschen Volkes.

Find an assignment M^* in G such that $c(M^*) \leq c(M)$ for all assignments M in G. From an algorithmic point of view we are faced with two problems

- to *characterize* optimal assignments,
- to *construct* optimal assignments.

The classical AP-algorithms use the complementarity conditions associated with the LP-formulation of the AP to prove optimality. In addition they are based on two construction principles: they either successively build up

- negative alternating cycles or
- shortest augmenting paths.

The approach we develop here does not use LP duality to verify optimality for an assignment. It is based on the following idea:

Let C' be a cost-matrix with $c'_{ij} \leq c_{ij}$ for all $(i, j) \in E$. If an assignment M can be shown to be optimal with respect to C' and if $c'(M) = c(M)$, then M is also optimal with respect to C. Note that the characterization of optimality by complementary slackness is also of this form, where C' is taken to be $c'_{ij} = u_i + v_j$. The crucial point is to construct a C' such that the optimality of M can easily be checked.

Our investigation was motivated by results which we had obtained for cardinality matching problems (cf. Derigs et al. [2]) and it was also guided by a classic observation due to Monge [5]. Monge could show for the slightly more general transportation problem that for a class of specially structured cost-matrices an optimal transportation solution can be obtained by applying the greedy-like north-west-corner rule. Those matrices are said to have the Monge-property. Note that applying this construction to an assignment problem gives the assignment $M = \{(1, 1), \ldots, (n, n)\}$.

The scope of this paper is not to propose a fast assignment algorithm, but to present a new simple, purely combinatorial optimality criterion and its algorithmic aspects.

In Section 2 we introduce the basic concept of Monge sequences and show in Section 3 how this concept can be used to construct an optimal assignment.

In Sections 4 and 5 we relate our approach to classical algorithms.

2. Monge sequences

Given a graph $G = (V, E)$ and a subset $E' \subseteq E$ we denote by $G \setminus E'$ the graph obtained by deleting $V(E')$, the set of nodes incident with edges in E', and all edges incident with nodes in $V(E')$.

An edge $\{u, v\}$ of a bipartite graph has the *Monge property* (with respect to a given cost-matrix $C = (c_{ij})$ if

$$c_{uv} + c_{rs} \leq c_{us} + c_{rv} \quad \text{for all } r \in V_1, s \in V_2.$$

A sequence $F = (e_1, e_2, \ldots, e_k)$ is called a *Monge sequence* if e_i has the Monge property in $G \setminus \{e_1, \ldots, e_{i-1}\}$ for $1 \leq i \leq k$.

Lemma 2.1. *Let $F=(e_1, e_2, \ldots, e_k)$ be a Monge sequence of a complete bipartite graph with respect to a given cost-matrix C. Then there exists an optimal assignment containing $e_1, \ldots, e_k$.*

Proof. Obviously, $M=\{e_1, \ldots, e_k\}$ is a matching in G. Let M' be any perfect matching with $M \subseteq M'$ in G and assume that there exists a negative cycle Q with respect to M' which contains an element e_i but, by induction, none of the elements e_j, $j<i$. Let $e_i=\{u, v\}$ and let r and s be the neighbours of u and v in Q. Since e_i has the Monge property, i.e., $c_{rs} \leq c_{us} - c_{uv} + c_{rv}$, there exists an alternating cycle Q' not containing e_i with $l(Q') \leq l(Q)$. □

Hence we have the following

Theorem 2.2. *Let $G=(V_1, V_2, E)$ be a complete bipartite graph. If $F=(e_1, e_2, \ldots, e_n)$ is a Monge sequence, then $M=\{e_1, \ldots, e_n\}$ is an optimal assignment.* □

As already mentioned in the introduction our approach is based on the following idea:

Given a cost-matrix $C'=(c'_{ij})$ with $c'_{ij} \leq c_{ij}$ for $(i, j) \in E$ and $F=(e_1, \ldots, e_n)$ a Monge sequence with respect to C', then $M=\{e_1, \ldots, e_n\}$ is an optimal assignment with respect to the cost-matrix $C=(c_{ij})$, if $c'_{ij}=c_{ij}$ for all $(i, j) \in M$.

3. Constructing Monge sequences

In this section we show how the concept of Monge sequences can be used for finding optimal assignments in bipartite graphs.

Consider the assignment $M=\{(1, 1), (2, 2), \ldots, (n, n)\}$ and define recursively a sequence of cost matrices $C=C^{(0)}, C^{(1)}, C^{(2)}, \ldots, C^{(n-1)}$ in the following way $(1 \leq k \leq n-1)$:

$$c_{ij}^{(k)} = \begin{cases} c_{ij}^{(k-1)} & \text{if } i \leq k \text{ or } j \leq k \\ \min\{c_{ij}^{(k-1)}, c_{ik}^{(k-1)} + c_{kj}^{(k-1)} - c_{kk}^{(k-1)}\} & \text{otherwise.} \end{cases}$$

We call this operation a *pivot-step* and we say that $C^{(k)}$ is obtained from $C^{(k-1)}$ by *pivoting* on element (edge) (k, k).

Then we have the following

Theorem 3.1. *M is an optimal assignment if and only if $c_{ii}^{(0)}=c_{ii}^{(n-1)}$ for $1 \leq i \leq n$.*

Proof. Suppose $c_{ii}^{(0)}=c_{ii}^{(n-1)}$ for $1 \leq i \leq n$. Then (1, 1) has the Monge property in $C^{(1)}$ and hence in each succeeding $C^{(k)}$. Similarly, (1, 1), (2, 2) is a Monge sequence in $C^{(2)}$ and in each succeeding $C^{(k)}$. Continuing this way, we see that $M=(1, 1)$,

$(2, 2), \ldots, (n, n)$ is a Monge sequence with respect to $C^{(n-1)}$. Thus, by Lemma 2.1, M is an optimal assignment with respect to the cost-matrix $C^{(n-1)}$. Since $c_{ij}^{(n-1)} \leq c_{ij}^{(0)}$ for all i, j it is clear that M is also optimal with respect to $C^{(0)}$.

Conversely, suppose $c_{ii}^{(n-1)} < c_{ii}^{(0)}$ for some $1 \leq i \leq n$. We need to show that M is not optimal. Let k be the first step so that $c_{ll}^{(k)} < c_{ll}^{(0)}$ for some l (necessarily $> k$). By relabelling if necessary we suppose $l = k+1$. We will show that there is a better matching of the first $k+1$ nodes than $(1, 1) \cdots (k+1, k+1)$, which shows that M is not optimal. Without loss of generality, we take $k = n-1$. Thus $c_{ii}^{(n-1)} = c_{ii}^{(0)}$ for $1 \leq i \leq n-1$ and $c_{nn}^{(n-1)} < c_{nn}^{(0)}$.

Now let $A^{(k)} = \{(i, j) : c_{ij}^{(k)} < c_{ij}^{(k-1)}\}$. We then inductively define matchings $M = M^{(n-1)}, \ldots, M^{(0)}$ such that

$$c^{(n-1)}(M^{(n-1)}) = c^{(n-2)}(M^{(n-2)}) = \cdots = c^{(0)}(M^{(0)}).$$

This then shows that M is not optimal since $c(M^{(0)}) = c^{(n-1)}(M) < c(M)$.

If $M^{(n-1)}, \ldots, M^{(k)}$ are already defined, then we consider three cases:

Case I: $M^{(k)} \cap A^{(k)} = \emptyset$. Let $M^{(k-1)} = M^{(k)}$. Then clearly $c^{(k-1)}(M^{(k-1)}) = c^{(k)}(M^{(k)})$

Case II: $M^{(k)} \cap A^{(k)} = \{(p, q)\}$. Set

$$M^{(k-1)} = M^{(k)} - \{(p, q), (k, k)\} \cup \{(p, k), (k, q)\}.$$

Then obviously

$$\begin{aligned} c^{(k-1)}(M^{(k-1)}) &= c^{(k-1)}(M^{(k)}) - c_{pq}^{(k-1)} - c_{kk}^{(k-1)} + c_{pk}^{(k-1)} + c_{kq}^{(k-1)} \\ &= c^{(k-1)}(M^{(k)}) - c_{pq}^{(k-1)} + c_{pq}^{(k)} \\ &= c^{(k)}(M^{(k)}). \end{aligned}$$

Case III: $|M^{(k)} \cap A^{(k)}| \geq 2$. Suppose k is the first index starting form $n-1$ where this case occurs, i.e., there exist

$$(p, q), (\bar{p}, \bar{q}) \in A^{(k)} \cap M^{(k)} \quad \text{with } p \neq \bar{p}, q \neq \bar{q}.$$

Then

$$c_{pq}^{(k)} < c_{pq}^{(0)} \quad \text{and} \quad c_{\bar{p}\bar{q}}^{(k)} < c_{\bar{p}\bar{q}}^{(0)}, \tag{1}$$

$$c_{pq}^{(k)} + c_{\bar{p}\bar{q}}^{(k)} \geq c_{p\bar{q}}^{(k)} + c_{\bar{p}q}^{(k)}. \tag{2}$$

(1) is clear, since $c_{pq}^{(k)} < c_{pq}^{(k-1)} \leq c_{pq}^{(0)}$ and the same for $(\bar{p}, \bar{q})$. (2) follows from:

$$\begin{aligned} c_{pq}^{(k)} + c_{\bar{p}\bar{q}}^{(k)} &= c_{kq}^{(k-1)} + c_{pk}^{(k-1)} - c_{kk}^{(k-1)} + c_{k\bar{q}}^{(k-1)} + c_{\bar{p}k}^{(k-1)} - c_{kk}^{(k-1)} \\ &\geq c_{pq}^{(k)} + c_{p\bar{q}}^{(k)}. \end{aligned}$$

We now claim that if for some $l \geq k$ there exists a pair $(p, q), (\bar{p}, \bar{q}) \in M^{(l)}$ such that (1) and (2) hold, then there also exists such a pair in $M^{(l+1)}$. This will lead to a contradiction, since by assumption $c_{ii}^{(n-1)} = c_{ii}^{(0)}$ for $1 \leq i \leq n-1$.

To prove our claim we distinguish three cases:

(i) $(p, q), (\bar{p}, \bar{q}) \in M^{(l+1)}$. Then $(p, q), (\bar{p}, \bar{q}) \notin A^{(l+1)}$, hence $c_{pq}^{(l+1)} = c_{pq}^{(l)}, c_{\bar{p}\bar{q}}^{(l+1)} = c_{\bar{p}\bar{q}}^{(l)}$ and (1) and (2) follow easily.

(ii) $(p, q), (\bar{p}, \bar{q}) \notin M^{(l+1)}$. Then by the choice of k we must have

$$M^{(l)} = M^{(l+1)} \setminus \{(l+1, l+1), (p', q')\} \cup \{(p', l+1), (l+1, q')\}$$

for some $(p', q') \in A^{(l+1)} \cap M^{(l+1)}$. Therefore

$$\{(p, q), (\bar{p}, \bar{q})\} = \{(l+1, q'), (p', l+1)\}$$

and

$$\begin{aligned} c_{pq}^{(l)} + c_{\bar{p}\bar{q}}^{(l)} &= c_{l+1, q'}^{(l)} + c_{p', l+1}^{(l)} \\ &< c_{l+1, l+1}^{(l)} + c_{p', q'}^{(l)} \qquad \text{since } (p', q') \in A^{(l+1)} \\ &= c_{p\bar{q}}^{(l)} + c_{\bar{p}q}^{(l)} \qquad \text{contradicting (2).} \end{aligned}$$

(iii) $(p, q) \in M^{(l+1)}, (\bar{p}, \bar{q}) \notin M^{(l+1)}$. Again by the choice of k we must have that

$$(\bar{p}, \bar{q}) \in M^{(l)} - M^{(l+1)} = \{(p', l+1), (l+1, q')\}$$

for some $(p', q') \in A^{(l+1)} \cap M^{(l-1)}$.

Clearly, (p, q) and (p', q') are distinct edges, since $(\bar{p}, \bar{q})$ and (p, q) are both in $M^{(l)}$ and $(\bar{p}, \bar{q})$ is either $(p', l+1)$ or $(l+1, q')$. We assume w.l.o.g. $(\bar{p}, \bar{q}) = (p', l+1)$. We now verify (1) and (2) for (p, q) and (p', q'):

$$c_{pq}^{(l+1)} \leq c_{pq}^{(l)} < c_{pq}^{(0)} \quad \text{and} \quad c_{p'q'}^{(l+1)} < c_{p'q'}^{(l)} \leq c_{p'q'}^{(0)},$$

$$\begin{aligned} c_{pq}^{(l+1)} + c_{p'q'}^{(l+1)} &= c_{pq}^{(l)} + c_{p'l+1}^{(l)} + c_{l+1, q'}^{(l)} - c_{l+1, l+1}^{(l)} \\ &= c_{pq}^{(l)} + c_{\bar{p}\bar{q}}^{(l)} + c_{l+1, q'}^{(l)} - c_{l+1, l+1}^{(l)} \\ &\geq c_{p\bar{q}}^{(l)} + c_{\bar{p}q}^{(l)} + c_{l+1, q'}^{(l)} - c_{l+1, l+1}^{(l)} \\ &= c_{p, l+1}^{(l)} + c_{p'q}^{(l)} + c_{l+1, q'}^{(l)} - c_{l+1, l+1}^{(l)} \\ &\geq c_{p'q}^{(l+1)} + c_{pq'}^{(l+1)}. \qquad \square \end{aligned}$$

4. Relationship to other classical concepts

The concept described in Section 3 leads to several different strategies for solving assignment problems. We will outline here two strategies which mimic the two basic combinatorial approaches mentioned in the introduction. We will shortly describe these principles here to enable the classification of our approach.

Given a matching M a path (cycle) P in G is called *M-alternating* if the edges in P are alternately in M and not in M. A M-alternating path P is called *M-augmenting* if P connects two nodes which are not matched by M. For a M-augmenting path P resp. M-alternating cycle P we define $l(P) := c(P \setminus M) - c(M \cap P)$ the *marginal*

cost or *length* of P. Then reversing the role of matching and nonmatching edges along P leads to a new matching

$$M \oplus P := (M \setminus P) \cup (P \setminus M) \quad \text{with } c(M \oplus P) = c(M) + l(P).$$

Now a matching is called *extreme* if it does not allow any negative M-alternating cycle, i.e., no alternating cycle P with $l(P) < 0$.

Now the negative *alternating cycle approach* is based on the fact that an assignment is optimal iff it is extreme. Then, starting from any assignment one successively improves a current assignment via negative alternating cycles until an *extreme* assignment is obtained.

The *shortest augmenting path approach* uses the following relationship. Let M be any extreme matching and P_{ij} a *shortest* M-augmenting path connecting two unmatched nodes, i and j, then $(M \oplus P_{ij})$ is extreme, too. Now starting from any extreme matching M, $M = \emptyset$ for instance, one successively augments along shortest augmenting paths until an assignment is obtained.

4.1. The dimension expanding approach

Let $M = \{(i_1, j_1), \ldots, (i_k, j_k)\}$ be any extreme matching in G, i.e., a matching not allowing negative alternating cycles and define $G|_M$ to be the graph induced by the nodes which are matched under M. Then M is an optimal perfect matching in $G|_M$. Thus after pivoting on the edges contained in M (in any order) we obtain a cost-matrix $C^{(k)}$ with the property

$$c^{(k)}_{i_l, j_l} = c^{(0)}_{i_l, j_l} \quad \text{for } l = 1, \ldots, k$$

Now choose any edge $(r, s) \in E$ s.t. $M' := M \cup \{r, s\}$ is a matching. Then either $c^{(k)}_{r,s} = c^{(0)}_{r,s}$, in which case M' is an optimal assignment in $G|_{M \cup \{r,s\}}$ or if $c^{(k)}_{r,s} < c^{(0)}_{r,s}$, then by our backtracking procedure we can construct a matching $\tilde{M}$ in $G|_{M \cup \{r,s\}}$ with

$$c^{(0)}(\tilde{M}) = c^{(k)}(M) + c^{(k)}_{r,s} = c^{(0)}(M) + c^{(k)}_{r,s} = c^{(k)}(M').$$

Now we argue that $\tilde{M}$ is an optimal assignment in $G|_{M \cup \{r,s\}}$. This can be seen as follows:

In any case, M' is an optimal assignment in $G|_{M \cup \{r,s\}}$ with respect to the cost-matrix $C^{(k)}$, due to Theorem 3.1.

Now $c^{(0)}_{ij} \geq c^{(k)}_{ij}$ for all $1 \leq i, j \leq n$, hence from the relation $c^{(0)}(\tilde{M}) = c^{(k)}(M')$ follows the optimality of $\tilde{M}$ in $G|_{\tilde{M}}$ with respect to the original cost-matrix $C^{(0)}$. That is, $\tilde{M}$ is extreme in G (with respect to C).

After at most n applications of our pivoting process and backtracking routine we obtain an optimal assignment M in G. Since every iteration is of complexity $O(n^3)$ we obtain a total complexity of $O(n^4)$. A closer look at this procedure shows, that at any iteration the set $(M' \setminus \tilde{M}) \cup (\tilde{M} \setminus M')$ is either an M'-alternating cycle or empty and hence $\tilde{M} = M \oplus P$ with P an M-augmenting path. Since M and $\tilde{M}$ are both

extreme, this path P is a shortest M-augmenting path connecting r and s. Hence this approach can be attributed as shortest augmenting path method.

4.2. The primal approach

In this variant we start from any assignment M in G and we keep pivoting on the elements contained in M (in any order) until for some element in M the corresponding cost coefficient is decreased. Then we use our backtracking routine starting from this element to improve the assignment and we restart the whole process. If $(n-1)$ pivot steps are possible without changing the cost coefficients for the edges contained in M, then M is an optimal assignment.

Clearly this procedure is finite, i.e., an optimal assignment M^* is obtained after a finite number of iterations since after each iteration the objective function value decreases by at least one unit.

Yet it is easy to show that this procedure runs in polynomial time if once a certain 'pivoting order' is chosen this order is maintained throughout the whole process.

To simplify our argumentation we assume w.l.o.g. $M=\{(1,1),\dots,(n,n)\}$ and that we choose the elements for pivoting in this 'natural' order. Now let $1\le k\le n-1$

$$c_{l,l}^{(k)}=c_{l,l}^{(0)} \quad \text{for } 1\le l\le k,$$

$$c_{k+1,k+1}^{(k)}<c_{k+1,k+1}^{(0)}.$$

Then $\tilde{M}=\{(1,1),\dots,(k,k)\}$ is extreme and our backtracking procedure will produce an improved assignment $M'=\{(1,j_1),\dots,(k+1,j_{k+1}),(k+2,k+2),\dots,(n,n)\}$ with $\tilde{M}':=\{(1,j_1),\dots,(k+1,j_{k+1})\}\subseteq M'$ optimal in $G|_{\tilde{M}\cup\{(k+1,k+1)\}}$.

Hence if in the next iteration we start pivoting with the elements from $\tilde{M}'$, in any order no cost-coefficient for elements in $\tilde{M}'$ will be altered. Thus after at most $O(n)$ iterations we end up with an optimal assignment. Note that the primal approach with this rule is exactly the dimension expanding approach with $(r,s)=(k,k)$ at each stage.

5. Monge sequences and the triple algorithm

In this section we shortly outline the relationship between our approach and the well-known triple algorithm (Floyd [3]) for solving shortest path problems in (directed) graphs, respectively for detecting negative cycles. Let $G=(V,E)$ be a directed graph. Given a matrix $C^{(0)}=(c_{ij}^{(0)})$ with $c_{ij}^{(0)}$ the length of the directed edge (i,j) for $i\neq j$ and $c_{ii}^{(0)}=0$, the triple algorithm for calculating the shortest path lengths between all pairs i and j runs as follows:

```
for k = 1, ..., n
  for i = 1, ..., n
    for j = 1, ..., n
      c_ij^(k) = min{c_ij^(k-1), c_ik^(k-1) + c_kj^(k-1)}
```

then $C^{(n)}=(c_{ij}^{(n)})$ gives the length of the shortest paths and a negative cycle exists if and only if $c_{ii}^{(m)}<0$ for some $i=1,\ldots,n$ and some $m=1,\ldots,n$.

Here $c_{ij}^{(k)}$ can be viewed as the length of the shortest path from i to j which does not contain the nodes $k+1,\ldots,n$ (except for the endnodes i and j).

Now let M be an assignment in a bipartite graph $G=(V_1, V_2, E)$. Then G contains a negative M-alternating cycle if and only if there exists an edge $(i,j)\in M$ such that for the shortest M-alternating path P_{ij} connecting i and j we have

$$l(P_{ij})<c_{ij}.$$

Now let w.l.o.g. $M=\{(1,1),\ldots,(n,n)\}$ and $C^{(0)}=C$. Then for any $(i,j)\in E\setminus M$, $c_{ij}^{(0)}$ is the length of the shortest M-alternating path P_{ij} connecting i and j not containing any matching edge.

Now for any $(i,j)\in E\setminus M$

$$c_{ij}^{(1)}:=\min\{c_{i1}^{(0)}+c_{1j}^{(0)}-c_{11}^{(0)}, c_{ij}^{(0)}\} \quad \text{for } 2\le i,j\le n$$

is the length of the shortest M-alternating path P_{ij} connecting i and j and not containing any matching edges from $\{(2,2),\ldots,(n,n)\}$.

Then if

$$c_{r,r}^{(1)}<c_{r,r}^{(0)} \quad \text{for any } r>1,$$

the shortest M-alternating path $P_{r,r}$ connecting $r\in V_1$ and $r\in V_2$ is shorter than $c_{r,r}$, hence $P_{r,r}\cup\{(r,r)\}$ is a negative M-alternating cycle.

In general we see that for $1\le k<n$ and $k+1\le i,j\le n$ the value $c_{ij}^{(k)}$ gives the length of the shortest M-alternating path connecting i and j not containing any matching edges from $\{(k+1,k+1),\ldots,(n,n)\}$.

Hence our approach can also be viewed as a proper generalization of the triple-algorithm to detect negative M-alternating cycles.

Acknowledgement

We thank the referees who helped to improve the presentation of this paper.

References

[1] U. Derigs, Programming in networks and graphs - On the combinatorial background and near-equivalence of network flow and matching algorithms, Institute für Ökonometrie und Operations Research, Universität Bonn (1984).

[2] U. Derigs, O. Goecke and R. Schrader, Bisimplicial edges, Gaussian elimination and matchings in bipartite graphs. in: U. Pape, ed., Graph-theoretic Concepts in Computer Science 1984 (Trauner Verlag, Linz., 1984) 79-87.

[3] R. Floyd, Algorithm 97: Shortest path, Comm. ACM 5 (1962) 345.

[4] H. Kuhn, The Hungarian method for the assignment problem, Naval Res. Logist. Quart. 2 (1955) 83-97.

[5] G. Monge, Déblai et Remblai, Memoires de l'Academie des Sciences (1781).